

In the Drawings

Please amend Figures 13b, 16c, 17, 18, 19, 20, 23, 26, 35, and 55 as shown in red in the attached drawings.

REMARKS

Applicants respectfully request that the Examiner enter the amendments set forth above prior to examining the above-referenced application.

Applicant amend the specification and Figures 13b, 16c, 17, 18, 19, 20, 23, 26, 35, and 55 to correct typographical errors. Specifically, reference numeral 32 is a duplicate. Therefore Applicants replace reference numeral 32 with reference numeral 41 in both the specification and Figures 13b, 16c, 17, 18, 19, 20, 23, and 26. Applicants add reference numeral 41 to the connection between NMS 60 and the network device 540 in Figure 35. Reference numeral 838 is added to the input marked "Alt. Input from other EX CTS" in Figure 55. Both reference numeral 41 and reference numeral 838 are referred to in the specification and used in other figures to designate the same part of the invention. No new matter is added by these amendments.

In addition, Applicants amend Figure 55 to remove an extraneous line section to indicate the correct connection of the *output* 770 to the Alt. *output* to other EX CTS. Support for this amendment can be found throughout the specification, for example, on page 197, lines 29-32. In particular, the specification recites that "the output 770 (marked "Alt. Output to other EX CTS") of timing module 76 may be provided to the other EX CTS and received as input 838 (marked "Alt. Input from other EX CTS"). Thus, no new matter is added by this amendment.

Slave SRMs 37a-37n then download and execute the device driver executable files (DD.exe) 56a-56n from memory 40. As one example, one port device driver 43a-43d may be started for each port 44a-44d on line card 16a. The port driver and port are linked together through the assigned port PID number.

5

In order to understand the significance of the PMD file (i.e., metadata), note that the MCD software does not have knowledge of board types built into it. Instead, the MCD parameterizes its operations on a particular board by looking up the card type and version number in the PMD file and acting accordingly. Consequently, the MCD software does not need to be modified, rebuilt, tested and distributed with new hardware. The changes required in the software system infrastructure to support new hardware are simpler modify logical model 280 (Fig. 3a) to include: a new entry in the PMD file (or a new PMD file) and, where necessary, new device drivers and applications. Because the MCD software, which resides in the kernel, will not need to be modified, the new applications and device drivers and the new DDL files (reflecting the new PMD file) for the configuration database and NMS database are downloaded and upgraded (as described below) without re-booting the computer system.

10

15

Network Management System (NMS):

Referring to Fig. 13b, as described above, a user / network administrator of computer system 10 works with network management system (NMS) software 60 to configure computer system 10. In the embodiment described below, NMS 60 runs on a personal computer or workstation 62 and communicates with central processor 12 over Ethernet network ⁴¹~~32~~ (out-of-band). Instead, the NMS may communicate with central processor 12 over data path 34 (Fig. 1, in-band). Alternatively (or in addition as a back-up communication port), a user may communicate with computer system 10 through a console interface / terminal (840, Fig. 2a) connected to a serial line 66 connecting to the data or control path using a command line interface (CLI) protocol. Instead, NMS 60 could run directly on computer system 10 provided computer system 10 has an input mechanism for the user.

25

30

During installation, an NMS database 61 is established on, for example, work-station 62 using a DDL executable file corresponding to the NMS database. The DDL file may be

downloaded from persistent storage 21 in computer system 10 or supplied separately with other NMS programs as part of an NMS installation kit. The NMS database mirrors the configuration database through an active query feature (described below). In one embodiment, the NMS database is an Oracle database from Oracle Corporation in Boston,
5 Massachusetts.

The NMS and central processor 12 pass control and data over Ethernet ⁴¹ using, for example, the Java Database Connectivity (JDBC) protocol. Use of the JDBC protocol allows the NMS to communicate with the configuration database in the same manner that it
10 communicates with its own internal storage mechanisms, including the NMS database. Changes made to the configuration database are passed to the NMS database to ensure that both databases store the same data. This synchronization process is much more efficient, less error-prone and timely than older methods that require the NMS to periodically poll the network device to determine whether configuration changes have been made. In these
15 systems, NMS polling is unnecessary and wasteful if the configuration has not been changed. Additionally, if a configuration change is made through some other means, for example, a command line interface, and not through the NMS, the NMS will not be updated until the next poll, and if the network device crashes prior to the NMS poll, then the configuration change will be lost. In computer system 10, however, command line
20 interface changes made to configuration database 42 are passed immediately to the NMS database through the active query feature ensuring that the NMS, through both the configuration database and NMS database, is immediately aware of any configuration changes.

25 Asynchronously Providing Network Device Management Data:

Typically, work-station 62 (Fig. 13b) is coupled to many network computer systems, and NMS 60 is used to configure and manage each of these systems. In addition to configuring each system, the NMS also interprets management data gathered by each system relevant to each system's network accounting data, statistics, security and fault logging (or some
30 portion thereof) and presents this to the user. In current systems, two distributed carefully synchronized processes are used to move data from a network system/device to the NMS. The processes are synchronized with each other by having one or both processes maintain the state of the other process. To avoid the problems associated with using two


Application No.: 09/669,861
Filed: September 26, 2000
Group Art Unit: Not Yet Assigned

For the Examiner's convenience, Applicants enclose a copy of pages 77 and 78 of the specification in which the above corrections are indicated in red.

The Examiner is urged to telephone the undersigned Attorney for Applicant in the event that such communication is deemed to expedite prosecution of this matter.

Respectfully submitted,

Date: Jan. 3, 2001



Lisa J. Michaud
Reg. No. 44,238
Attorney for Applicant(s)

Nutter, McClennen & Fish, LLP
One International Place
Boston, MA 02110-2699
Tel: (617)439-2550
Fax: (617)310-9550

FIG. 13b

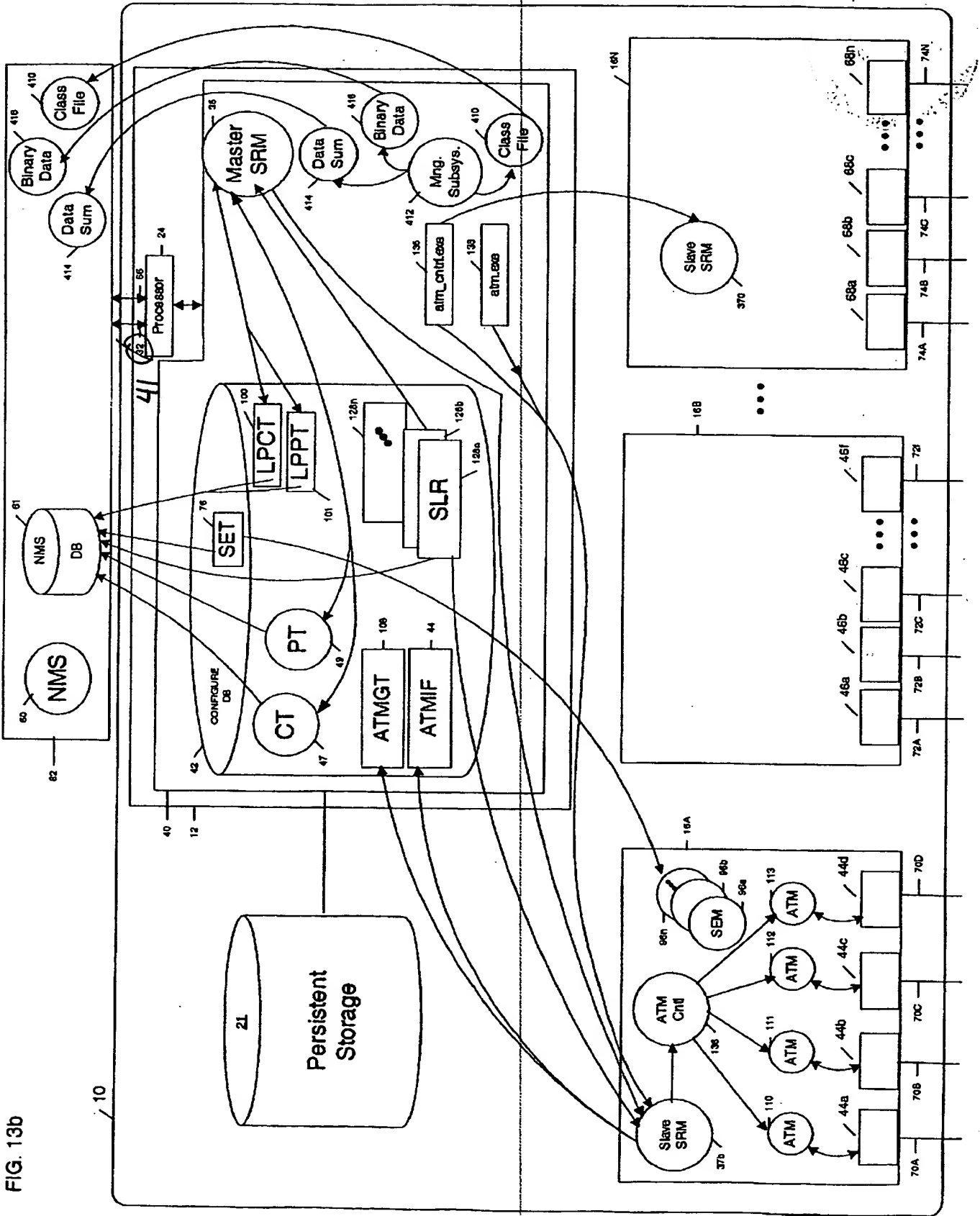


Figure 1 is a block diagram illustrating a network management system architecture. The system is organized into several interconnected components:

- Management Layer (Top):** Includes a **Processor 24** and **Config DB 42** (Configuration Database). The Processor 24 is connected to the Config DB 42 and a **Network Management System (NMS) 51**. The NMS 51 is further connected to a **Database (DB) 52** and a **Network Management System (NMS) 60**. A **Network Management System (NMS) 66** is also shown, connected to the Processor 24.
- Network Layer (Middle):** Consists of a **Network (10)** containing multiple **Network Service (NS) 220a, 220b, 220c, 220n** and **ATM 221a, 221b, 221n** nodes. These nodes are interconnected via a **Network 100**.
- Storage Layer (Bottom):** Includes **Persistent Storage 21**, which is connected to the Config DB 42 and the Network 100.
- Control Layer (Bottom):** Features a **Control 222** block, which is connected to the Network 100 and a **Control 44a** block. The Control 222 is also connected to a **Control 96a** block.

The diagram shows the flow of data and control signals between these components, highlighting the integration of network management, storage, and control functions.

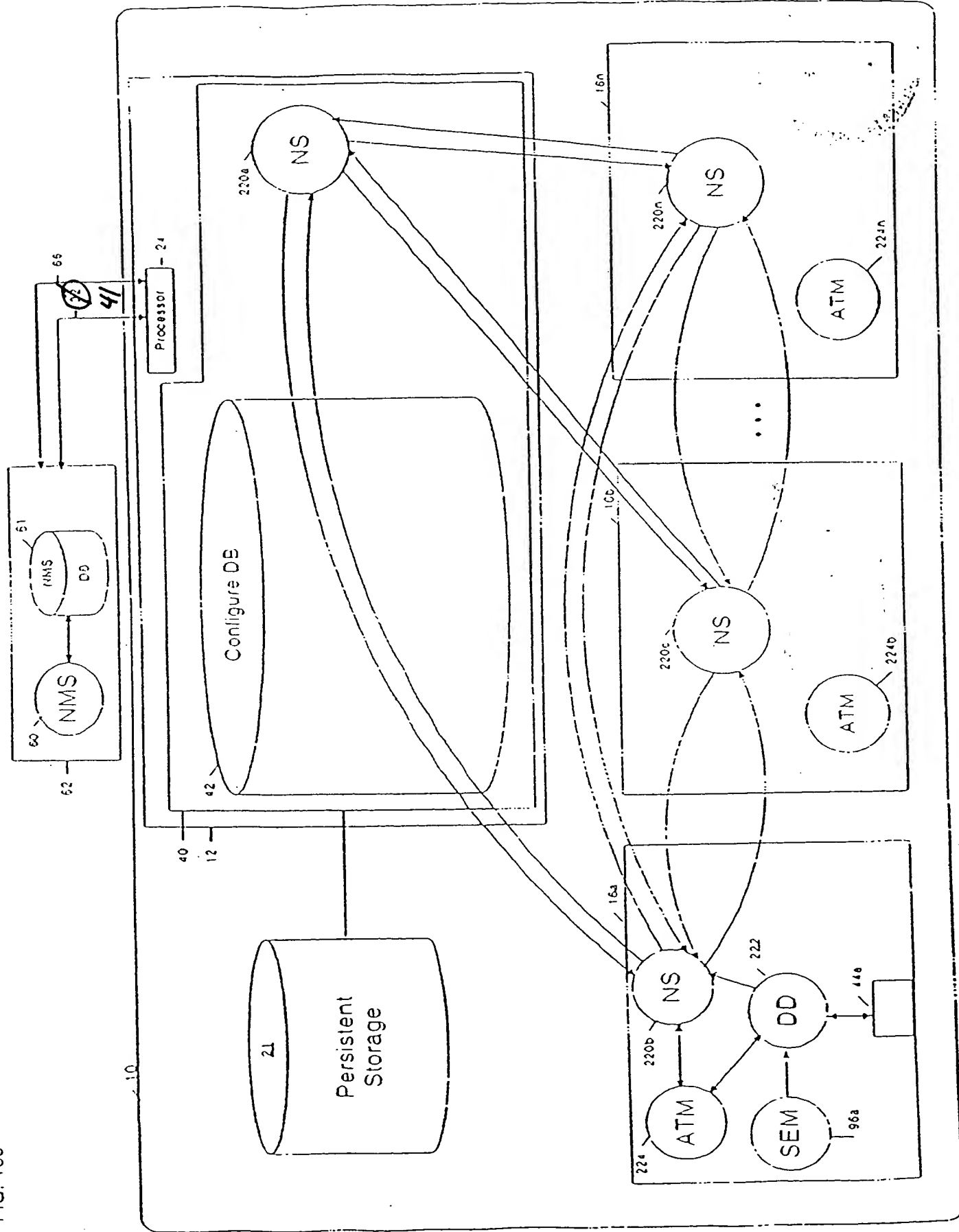


FIG. 17

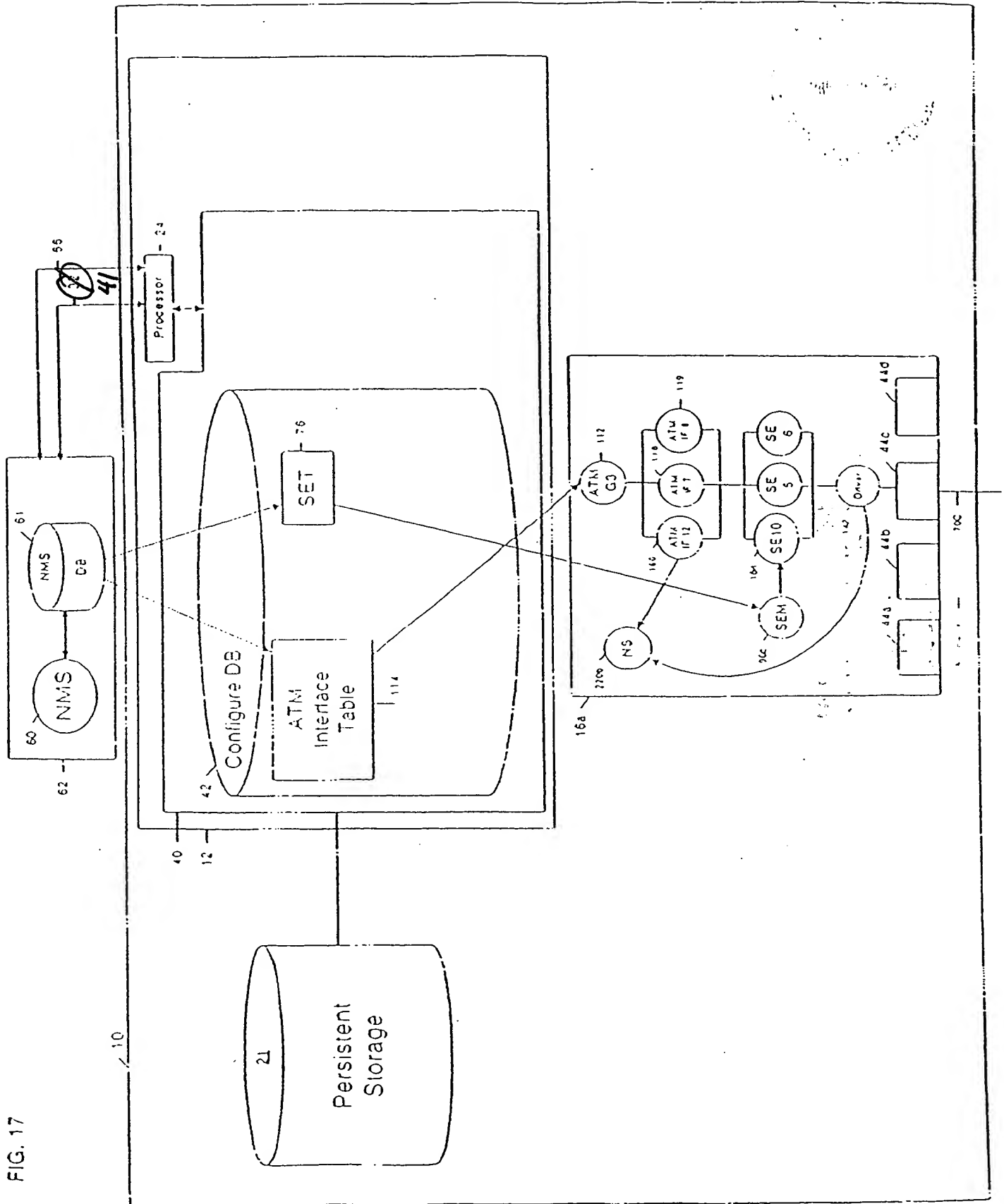


FIG. 18

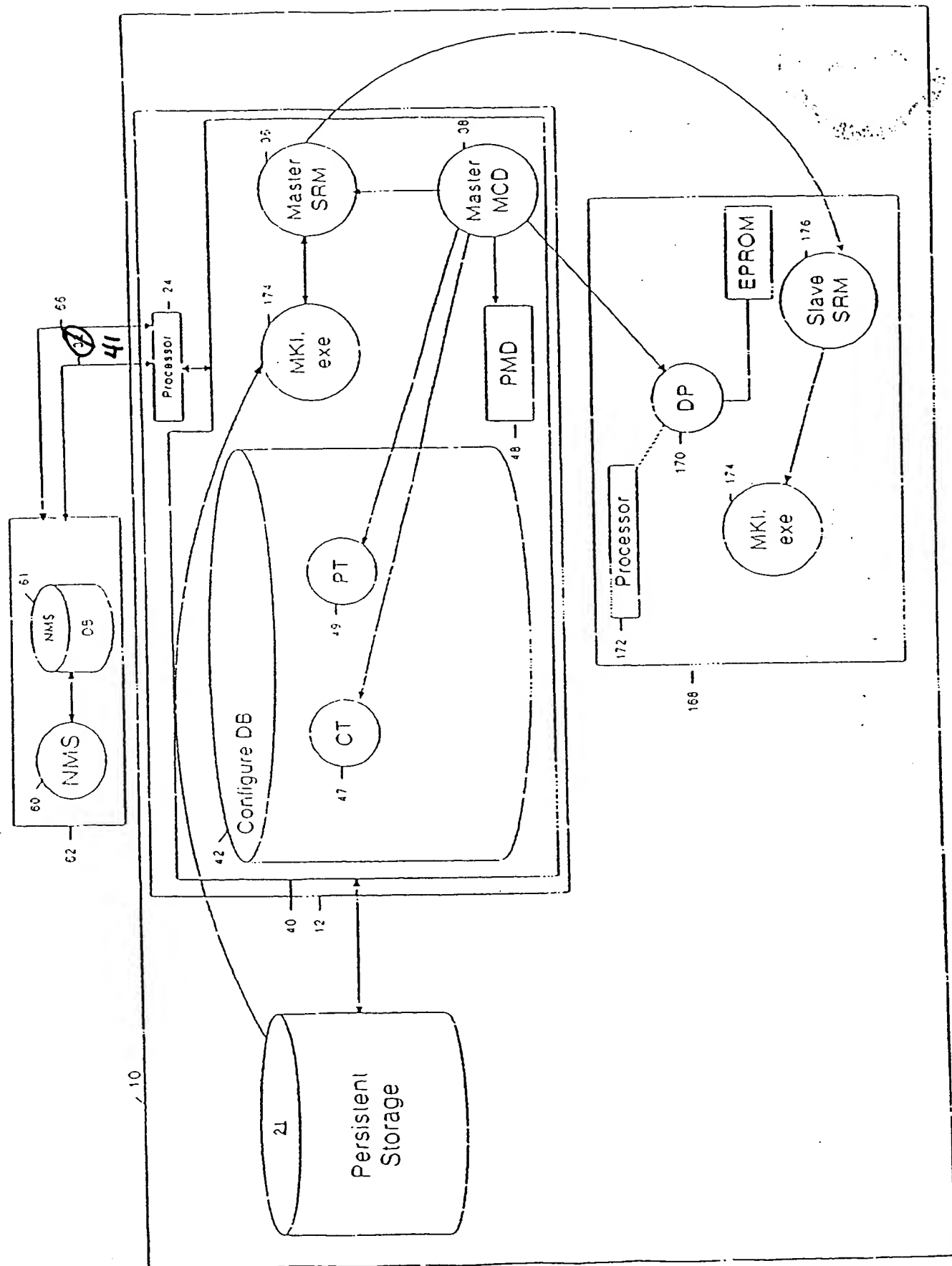
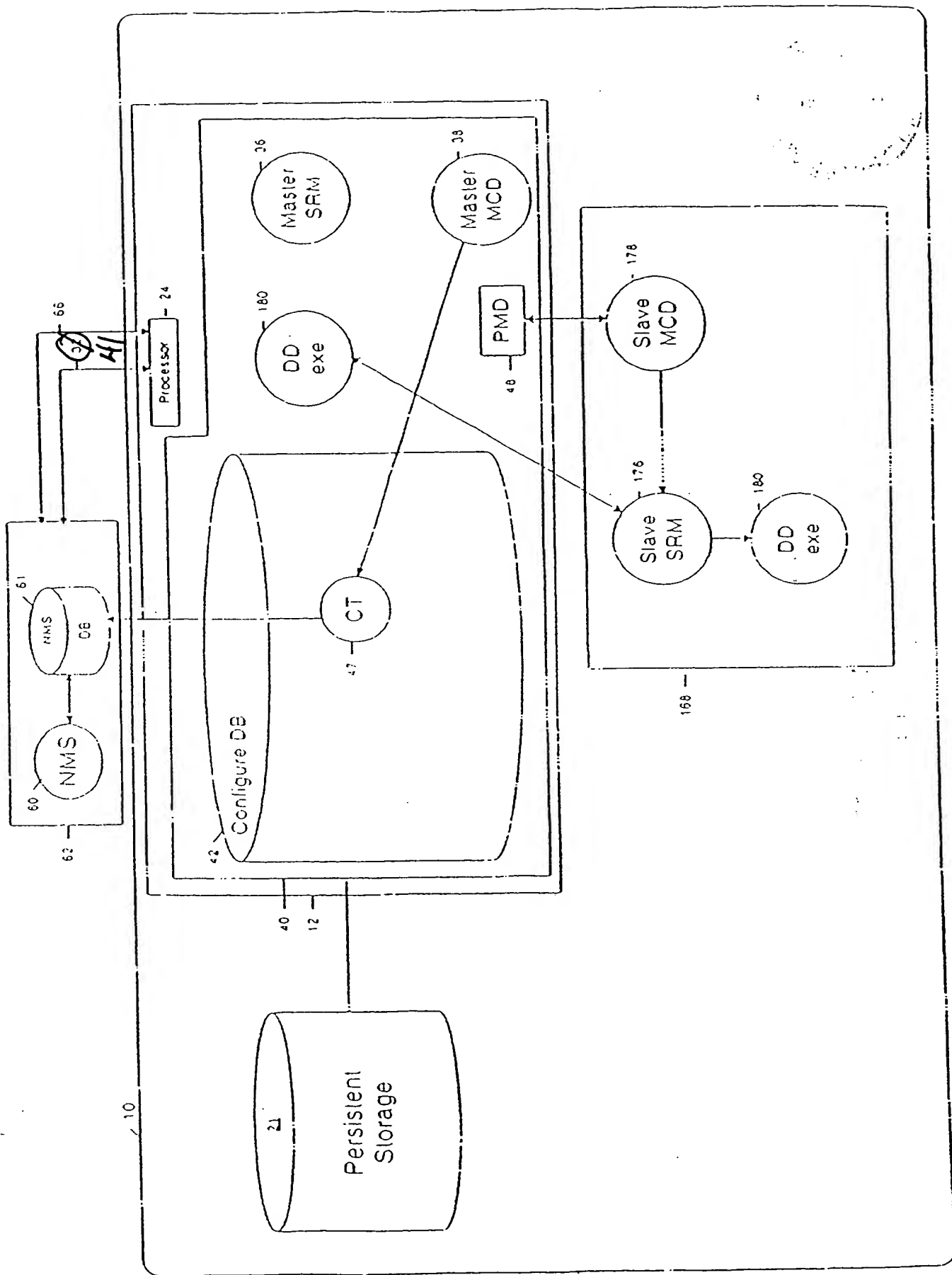


FIG. 19



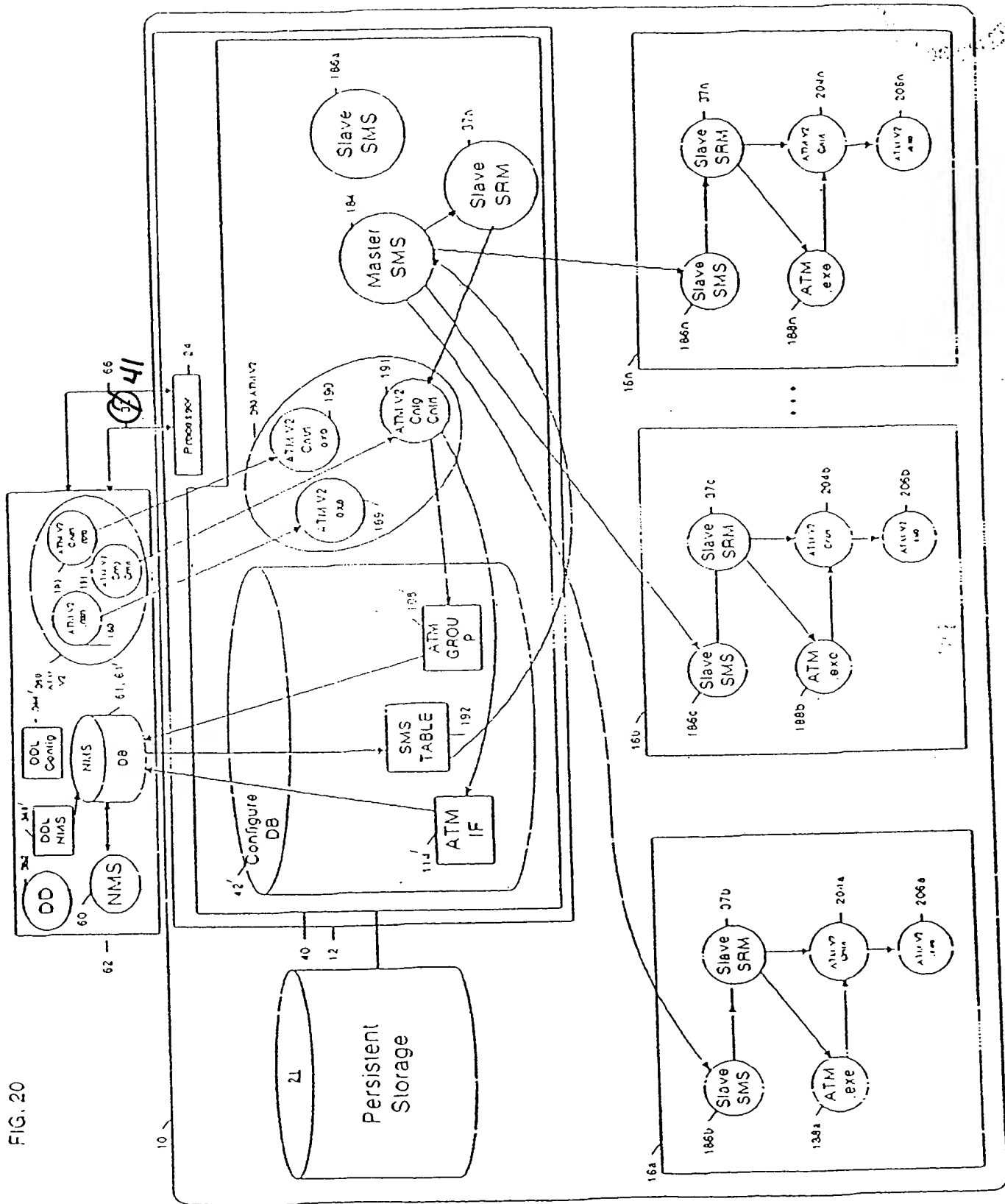
[illegible]

FIG. 23

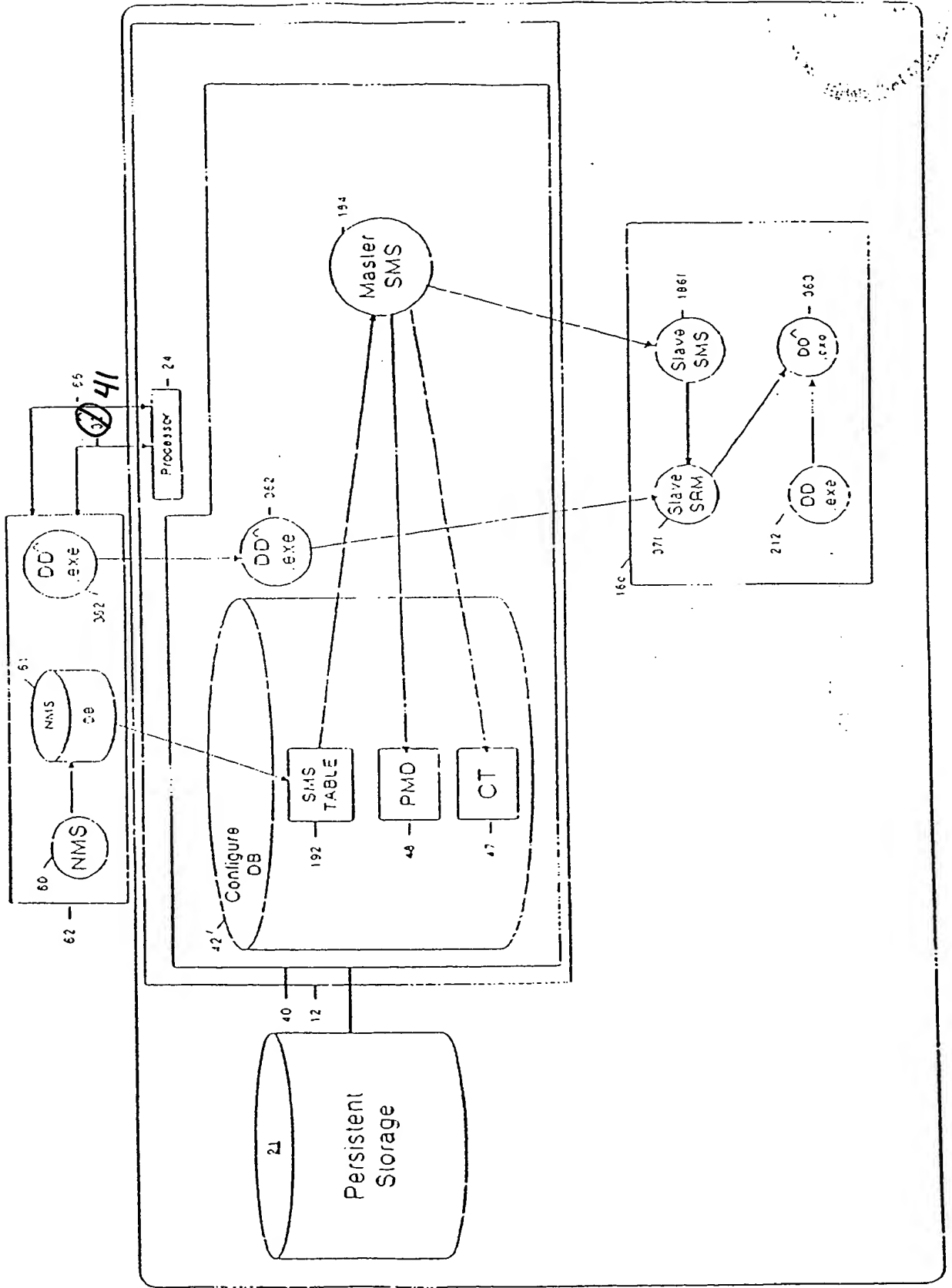


FIG. 26

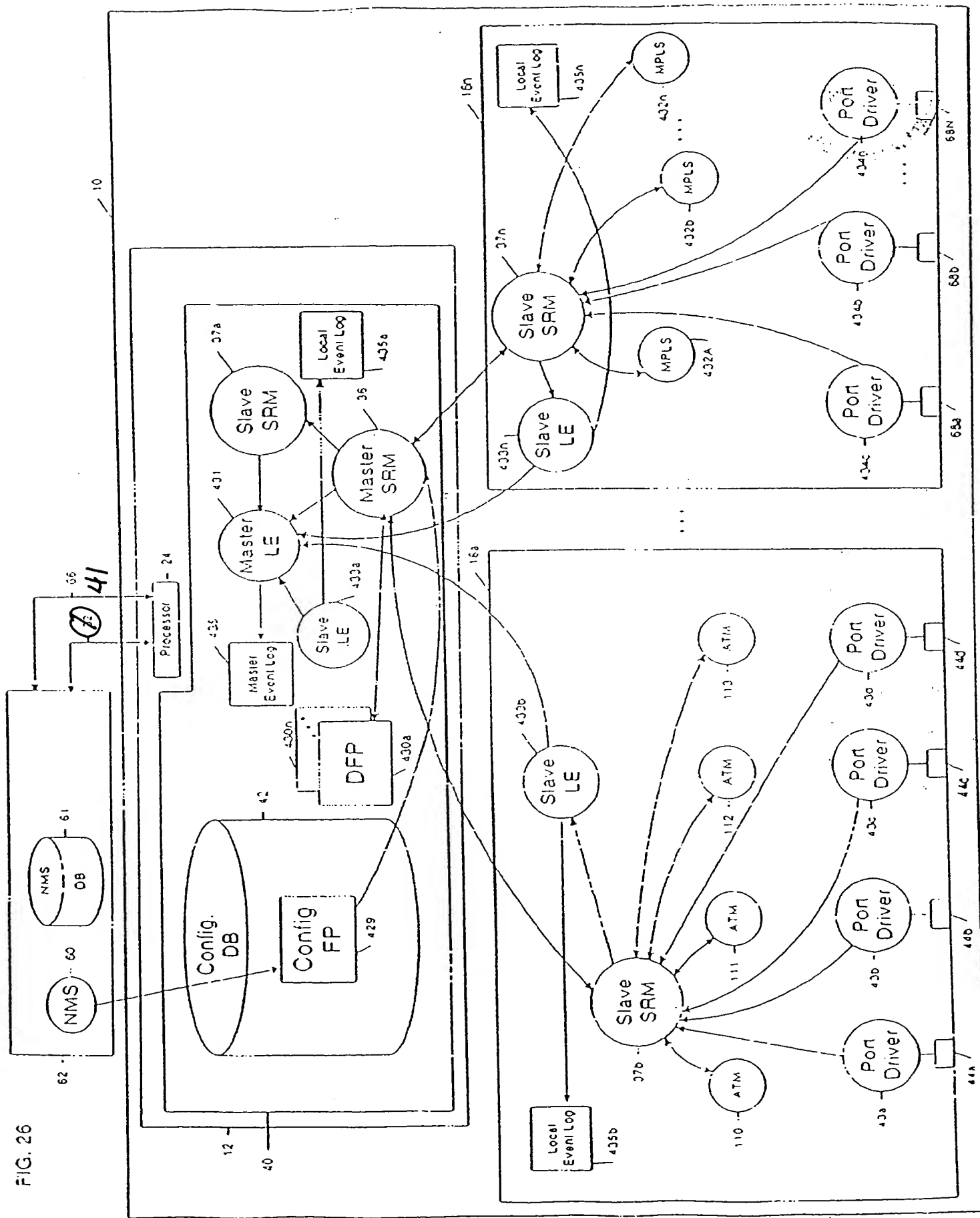
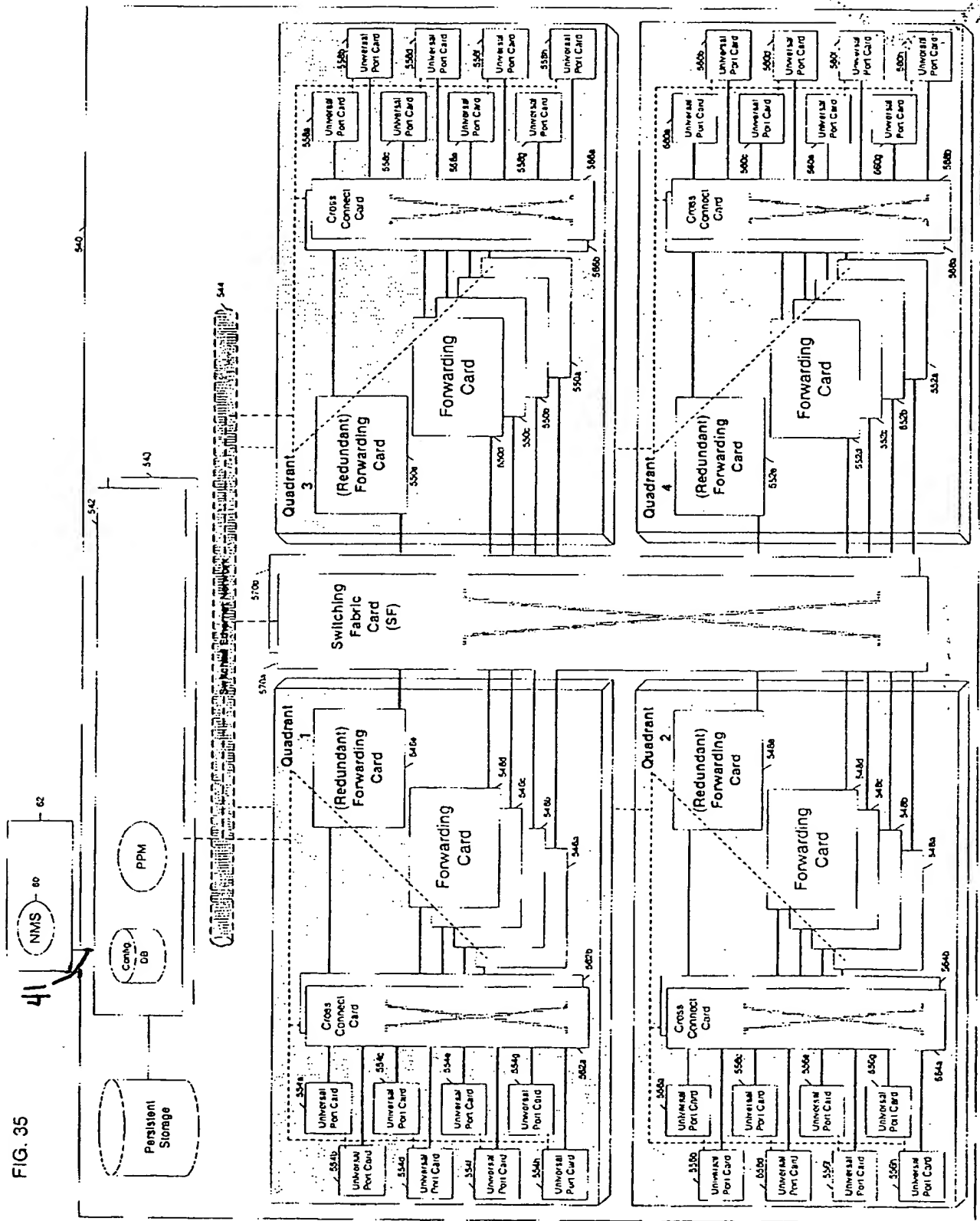


FIG. 35



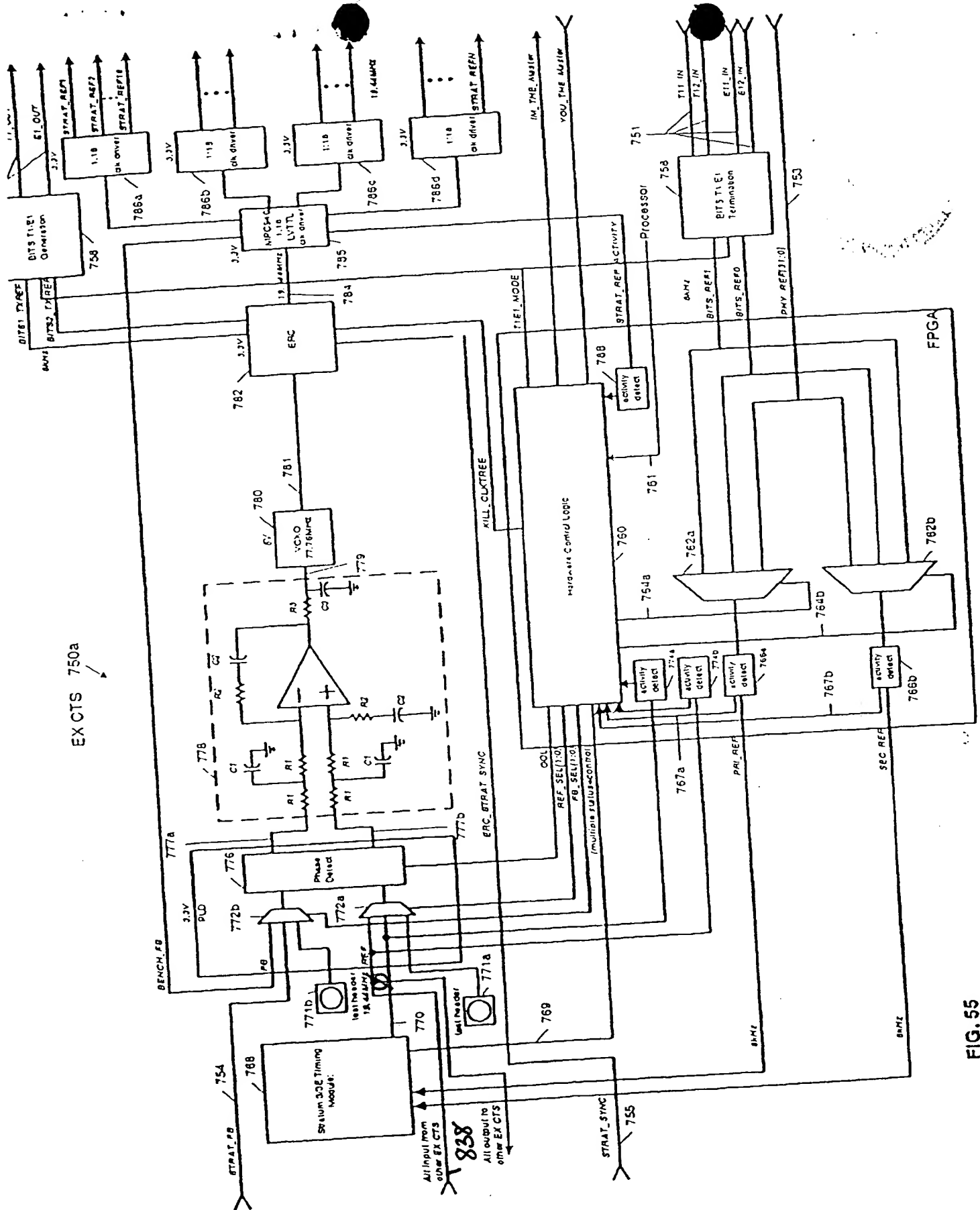


Fig. 55